IN THE CLAIMS

1. (Previously Presented) A method comprising:

providing prerequisite information regarding pages of a graphical user interface (GUI) that are prerequisites to other pages of the GUI, each page including one or more sub-components;

in response to a request to display a destination page and with reference to the prerequisite information, instantiating a container object corresponding to the requested destination page of the GUI;

the container object identifying one or more prerequisite pages associated with the destination page by invoking an execution of a method of the container object;

the container object determining which sub-component of the one or more sub-components of an identified prerequisite page is a decider sub-component that is capable of confirming whether or not requirements of the identified prerequisite page have been satisfied;

the container object determining whether or not the requirements of the identified prerequisite page have been satisfied by invoking a method of a component object corresponding to an instance of the decider sub-component that causes stored information regarding the state of the identified prerequisite page to be retrieved from a current component object, the component object being a child object of the container object; and

causing the output of the destination page to be displayed if all the requirements of the one or more identified prerequisite pages have been satisfied, otherwise causing the output of a prerequisite page of the one or more identified prerequisite pages having one or more requirements that have not been satisfied to be displayed by invoking an execution of a method of the corresponding component object to stream the content to the container object.

2. (Previously Presented) The method of claim 1 wherein the prerequisite information is stored in a Java properties file, wherein the method further comprises:

the container object instantiating a page sub-component prerequisite object using information stored in the Java properties file; and

retrieving the prerequisite information by executing a method of the page sub-component prerequisite object, wherein the page sub-component prerequisite object streams the prerequisite information to the container object in response to the execution.

3. (Previously Presented) The method of claim 2, further comprising supporting hierarchical relationships of prerequisite pages by:

iterating through each of the identified prerequisite pages associated with the destination page in a predetermined order by executing a method of the corresponding page sub-component prerequisite object until encountering the first prerequisite page that has one or more requirements that have not been satisfied; and

displaying the first prerequisite page of the identified prerequisite pages before displaying a second prerequisite page of the identified prerequisite pages that has one or more requirements that have not been satisfied, the second prerequisite page being dependent upon the first prerequisite page according to the predetermined order.

4. (Original) The method of claim 1, wherein the request to display the destination page comprises a HyperText Transfer Protocol (HTTP) request, and wherein the pages of the GUI comprise web pages.

5. (Previously Presented) The method of claim 2, wherein the prerequisite information includes a prerequisite property stored in the Java properties file for each of the pages of the GUI, the prerequisite property comprising a string identifying the one or more prerequisite pages.

6. (Previously Presented) The method of claim 5, wherein the prerequisite information in the Java properties file is structured as a list of attribute-value pairs, and wherein a syntax for identifying a first page, $page_1$, and a second page, $page_2$, as prerequisites of a third page, page 3, is substantially as follows:

$page_3$.prereq = $page_1$ $page_2$.

7. (Previously Presented) The method of claim 2, further comprising modifying the prerequisite information via the Java properties file without necessitating recompilation of software code corresponding to the component object.

8. (Previously Presented) The method of claim 1, wherein said determining whether or not the requirements of the identified prerequisite page have been satisfied includes requesting that a page prerequisite object verify whether all its requirements have been satisfied by invoking an execution of a method of the page sub-component prerequisite object.

9. (Previously Presented) The method of claim 1, wherein page objects corresponding to the pages of the GUI and page prerequisite objects responsible for ensuring satisfaction of one or more prerequisite conditions are loosely coupled and are dynamically associated with each other by way of the prerequisite information.

10. (Previously Presented) A graphical user interface (GUI) system for enforcing page prerequisites comprising:

a properties data store including information regarding pages of the GUI that are prerequisites to other pages of the GUI;

a base agent to respond to requests to display a destination page of the GUI, in response to a request to display the destination page, the base agent causing the output of the destination page to be displayed if all the requirements of one or more prerequisite pages associated with the destination page have been satisfied, otherwise causing the output of a prerequisite page of the one or more prerequisite pages to be displayed by invoking an execution of a method of an object corresponding to each page associated with the destination page; and

a prerequisite factory decoupling the pages from their respective prerequisite pages, the prerequisite factory to determine whether or not requirements of one or more identified prerequisite pages have been satisfied by causing information regarding the state of the one or more identified prerequisite pages to be retrieved from a current context.

11. (Previously Presented) The system of claim 10, wherein the prerequisite factory identifies the one or more prerequisite pages associated with the destination page by accessing the properties data store, determines which of one or more sub-components of an identified prerequisite page is a decider sub-component that is capable of confirming whether or not requirements of the identified prerequisite page have been satisfied, creates an instance of the decider sub-component, and determines whether or not the requirements of the identified prerequisite page have been satisfied by invoking a method of the instance that causes

information regarding the state of the identified prerequisite page to be retrieved from the current context.

12. (Previously Presented) The system of claim 10, wherein the properties data store is a Java properties file, wherein the base agent is to

instantiate a page sub-component prerequisite object using information stored in the Java properties file, and

execute a method of the page sub-component prerequisite object to cause the page sub-component prerequisite object to stream the prerequisite information to the base agent.

13. (Previously Presented) The system of claim 12, wherein the prerequisite factory supports hierarchical relationships of prerequisite pages by:

iterating through each of the identified prerequisite pages associated with the destination page in a predetermined order by executing a method of the corresponding page sub-component prerequisite object until encountering the first prerequisite page that has one or more requirements that have not been satisfied; and

displaying the first prerequisite page of the identified prerequisite pages before displaying a second prerequisite page of the identified prerequisite pages that has one or more requirements that have not been satisfied, the second prerequisite page being dependent upon the first prerequisite page according to the predetermined order.

14. (Original) The system of claim 10, wherein the requests correspond to HyperText Transfer Protocol (HTTP) requests, and wherein the pages of the GUI comprise web pages.

15. (Previously Presented) The system of claim 12, wherein the information of the properties data store includes a prerequisite property stored in the Java properties file for each of the pages of the GUI, the prerequisite property comprising a string identifying the one or more prerequisite pages.

16. (Previously Presented) The system of claim 12, wherein at least a portion of the information of the properties data store is structured as a list of attribute-value pairs stored in the Java properties file, and wherein a syntax for identifying a first page, $page_1$, and a second page, $page_2$, as prerequisites of a third page, $page_3$, is substantially as follows:

$page_3$.prereq + $page_1$ $page_2$.

17. (Previously Presented) The system of claim 10, wherein prerequisite relationships among two or more pages of the pages of the GUI are modified without necessitating recompilation of software code corresponding to the page sub-components by editing the information of the properties data store.

18. (Previously Presented) The system of claim 10, further comprising page objects corresponding to the pages of the GUI and page prerequisite objects responsible for ensuring satisfaction of one or more prerequisite conditions are loosely coupled and are dynamically associated with each other by way of the prerequisite information by invoking an execution of a method of the corresponding page sub-component prerequisite object.

19. (Previously Presented) A method comprising:

identifying, at run-time, one or more prerequisite web pages associated with a requested web page by accessing a properties file by invoking a method of a container object corresponding to the requested web page;

the container object determining whether requirements of the one or more identified prerequisite pages have been satisfied by instantiating and executing a method of one or more prerequisite objects corresponding to the identified prerequisite pages; and

causing the output of the requested web page to be displayed if all the requirements of the one or more identified prerequisite pages have been satisfied, otherwise causing the output of a prerequisite page of the one or more identified prerequisite pages having one or more unsatisfied requirements to be displayed by executing a method of an object corresponding to the prerequisite page.

20. (Previously Presented) A machine-readable medium having stored thereon data representing sequences of instructions, the sequences of instruction which, when executed by a processor, cause the processor to:

identify one or more prerequisite pages associated with a destination page by accessing a properties file in response to a request for the destination page by invoking a method of a container object corresponding to the destination page, the properties file including prerequisite information regarding pages of a graphical user interface (GUI) that are prerequisites to other pages of the GUI;

determine using the container object which sub-component of an identified prerequisite page is capable of confirming whether or not requirements of the identified prerequisite page have been satisfied;

determine whether the requirements of the identified prerequisite page have been satisfied by invoking from the container object a method of a component object corresponding to an instance of the sub-component that causes stored information regarding the state of the identified prerequisite page to be retrieved from a current component object, the component object being a child object of the container object; and

cause the output of the destination page to be displayed if all the requirements of the one or more identified prerequisite pages have been satisfied, otherwise cause the output of a prerequisite page of the one or more identified prerequisite pages having one or more requirements that have not been satisfied to be displayed by invoking an execution of a method of the corresponding component object to stream the content to the container object.

21. (Previously Presented) The method of claim 1, wherein at least one component object comprises one or more sub-container objects, and wherein at least one of the sub-container objects includes one or more sub-components.

22. (Previously Presented) The method of claim 21, wherein the container objects and the component objects are maintained in a hierarchical structure, wherein the component objects are implemented within one or more leaf nodes, and wherein the container objects are implemented within one or more non-leaf nodes, each non-leaf node having at least one of a leaf sub-node and a non-leaf sub-node.

23. (Previously Presented) The method of claim 22, wherein a container object represents a page and a component object represent content of the page including at least one of the following:

a chart;

a table;

a scroll list; and

a data entry.

24. (Previously Presented) The method of claim 5, wherein a first sub-component is identified as a prerequisite of a second sub-component by assigning the first sub-component to the second sub-component having the string identifying the prerequisites appended to the second sub-component separated by a delimiter.

25. (Previously Presented) The method of claim 24, wherein the prerequisites of the second sub-component is inherited from the first sub-component.

26. (Previously Presented) The method of claim 25, wherein at least one prerequisite of the second sub-component used in a context is overridden by assigning an overriding value to the second sub-component having an identity of the context prefixed to an identity of the second sub-component separated by a delimiter.